Engineering Analysis with Boundary Elements 96 (2018) 115–122

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

Adaptive moving knots meshless method for simulating time dependent partial differential equations

Qinjiao Gao[a], Zongmin Wu[b,c,*], Shenggang Zhang[d]

[a] School of Business, Dalian University of Foreign Languages, Dalian 116044, PR China
[b] Shanghai Key Laboratory for Contemporary Applied Mathematics, School of Mathematical Sciences, Fudan University, Shanghai 200433, PR China
[c] Shanghai Center for Mathematical Sciences, Shanghai 200433, PR China
[d] Department of Health Statistics, Dalian Medical University, Dalian 116044, PR China

ARTICLE INFO

MSC:
65D05
65D10
65K10
65N50

Keywords:
Moving knots
Equidistribution Principle
Meshless method
Time dependent PDEs

ABSTRACT

This paper presents an improvement of Huang's moving knots method (Huang et al.) for solving the PDE on the contours of Equidistribution Principle (EP-contours). Firstly, a new moving knots strategy is generated by discovering a significant factor ignored (dropped) by Huang (Huang et al.). The proposed strategy ensures that arbitrary initial knots could asymptotically converge to the EP-contours. Then the moving knots strategy and PDE are simulated by using multi-quadric (MQ) quasi-interpolation step by Step (3.3, 3.4). Error estimates of the algorithm are given. At last, numerical experiments are provided to illustrate the validity of the algorithm. Both theoretical analysis and numerical results show that the proposed algorithm benefits the methods in Huang et al. and Wu.

1. Introduction

Time dependent PDEs [1] are able to describe phenomena in various fields, such as engineering, physics, biology, economics, etc. Supported by the development of the computer, the numerical methods [2] have become efficient ways to study such problems. In this paper, we study the numerical solution of the well defined time dependent PDE

$$u_t(x,t) = \mathcal{L}(u)(x,t),\ \ x \in [a,b],\ \ t \in [0,T], \tag{1.1}$$

where $\mathcal{L}$ represents a differential operator involving only derivatives of u respect to space x (without loss of generality we take $[a,b]=[0,1]$ throughout the paper).

Firstly, Eq. (1.1) is discretized on time with forward divided difference method:

$$u(x,t_{k+1}) = u(x,t_k) + \Delta t_k \mathcal{L}(u)(x,t_k), \tag{1.2}$$

$\Delta t_k = t_{k+1} - t_k$ is the kth time step, then the resulting Eq. (1.2) will be discretized on space.

Generally, the space discretization methods could be divided into two categories: the Eulerian and the Lagrangian. In the Eulerian methods [3], the solutions are observed at fixed space locations, i.e., the PDE is simulated on fixed knots $\{x_j, t_k, u(x_j, t_k)\}$. When the solutions of original PDE involve large variations, such as boundary layers and even shock waves, the alternative Lagrangian methods [4,5] perform better in both efficiency and accuracy, since the knots x_j^k (the jth knot on time t_k), $j = 0,1,\cdots,N$ are moved with time and always concentrated in regions with large variations. For instance, it is appreciated if the knots distribute uniformly on the arc-length of solutions at fixed time t_k.

One key feature in developing moving knots algorithms lies in formulating a satisfactory strategy for moving knots. Wherein the EP [6,7] has turned out to be an excellent principle. Precisely, given a function $u(x, t)$, and by introducing a computational coordinate ξ, the one-to-one coordinate transformation is denoted by $x = x(\xi, t)$ with the boundary conditions $x(0,t) = 0$ and $x(1,t) = 1$. The EP-contour is defined to be the curve which is solved by the following EP (1.3) when ξ is a constant:

$$E(x(\xi,t),t) \doteq \int_0^{x(\xi,t)} M(\tilde{x},t)d\tilde{x} - \xi \int_0^1 M(\tilde{x},t)d\tilde{x} = 0, \tag{1.3}$$

where $M = \sqrt{1+u_x^2}$ is differentiation of the arc-length. Clearly, when $\xi_j = \frac{j}{N}, j = 0,1,\cdots,N$, the EP-contours $x_j(t) = x(\xi_j,t)$ are equidistributed on function $u(x, t)$.

As pointed out by Huang [5], a number of knots moving strategies were formulated for solving the PDE on the EP-contours. Tang et al. [8] developed the harmonic map between the original space and the computational space by the iteration procedure. Each iteration step was

* Corresponding author at: Shanghai Key Laboratory for Contemporary Applied Mathematics, School of Mathematical Sciences, Fudan University, Shanghai 200433, PR China.

E-mail addresses: 09110180026@fudan.edu.cn (Q. Gao), zmwu@fudan.edu.cn (Z. Wu), zsgdalian@mail.dlut.edu.cn (S. Zhang).

https://doi.org/10.1016/j.enganabound.2018.08.010
Received 25 December 2017; Received in revised form 13 August 2018; Accepted 16 August 2018

to move the knots closer to the harmonic map. Levesley et al. [9] proposed the adaptive algorithm which aimed to assess the local approximation quality and improve the error in the specified region. [10] presented the moving knots equation by which the knots were moved along the characteristic line of the original PDE. Huang et al. [7] constructed the moving mesh PDEs (MMPDEs), which targeted to move the knots closer to the EP-contours on each time step. It is worth mentioning that Huang's moving mesh methods were so popular in applications, that their codes were employed in the software MATLAB (bugersode.m).

For the knots moving strategies mentioned above, some [7,8,11,12] directly focus on moving the knots closer to the EP-contours on fixed time step. However, they neglect maintaining the (nearly) equidistribution during time iterations. Once the solutions vary steeply with time, the (nearly) equidistribution would be destroyed after each time iteration. Others [10,13] target to keep the initial distribution, yet they forget to move the knots closer to the EP-contours during each time iteration. Combining those two ideas, a new strategy for moving knots is constructed on the basis of EP. The strategy and the numerical solutions are estimated step by step:

$$\begin{cases} x_j^{k+1} = x_j^k - \frac{E(x_j^k,t_k)}{M(x_j^k,t_k)} - \Delta t_k \frac{\int_0^{x_j^k} M_t(\tilde{x},t_k)d\tilde{x} - \frac{j}{N}\int_0^1 M_t(\tilde{x},t_k)d\tilde{x}}{M(x_j^k,t_k)}; & (1.4) \\ u(x_j^{k+1},t_{k+1}) = u(x_j^{k+1},t_k) + \Delta t_k \mathcal{L}(u)(x_j^{k+1},t_k). & (1.5) \end{cases}$$

Early in 1994, Huang [7] has applied similar idea in constructing the MMPDEs. However, the terms including 'M_t' were dropped because of their difficulty to be calculated. Without these terms, a time lag [5] always exists, which makes the time step sizes of knots redistribution small in order to avoid instability and even knots tangling. The situation is worsen when the solutions propagate fast or vary largely according to time. To remedy this problem, various algorithms have been designed [5,14,15]. Their general idea is several sub-steps of knots redistributions companied with one step of PDE solving. These algorithms are inefficient in the sense that they spend most of the computing time on the knots redistributions.

In this paper, the time lag problem is settled by calculating out the term 'M_t' exactly: based on the original PDE (1.1), 'M_t' is calculated precisely by transforming the time derivatives into the space derivatives,

$$M_t = \frac{u_x u_{xt}}{\sqrt{1+u_x^2}} = \frac{u_x (u_t)_x}{\sqrt{1+u_x^2}} = \frac{u_x(\mathcal{L}(u))_x}{\sqrt{1+u_x^2}}. \tag{1.6}$$

When the terms including 'M_t' are added, the time lag disappears. As a result, the time step sizes of knots redistribution could be enlarged and the possibility of knots tangling could be reduced. It is proved that, arbitrary initial knots will asymptotically converge to the equidistributed case following the provided strategy. Therefore, it could simulate PDEs with faster propagating and steeper varying solutions. Moreover, the moving knots strategy is simple, robust and easy to program. Furthermore, the present strategy benefits both the methods in Huang [5,10].

In this paper, the schemes (1.4) and (1.5) are estimated applying the MQ quasi-interpolation meshless method [10,13,16–18]. Compared with the mesh based methods [8,19], which need a proper structure of the mesh, and require to solve the PDE after coordinate transformation, the MQ quasi-interpolation method gives an approximant directly without solving any large-scale linear system of equations. What's more, it requires no structure of the knots, hence we can move the knots more freely. In addition, the PDE is solved in the original coordinate and need no coordinate transformation. In conclusion, the proposed algorithm is simple, efficient and easy to implement. The error estimates of the proposed algorithm are given.

An outline of the paper is as follows. In Section 2, a new moving knots strategy is generated in Theorem 1 on the basis of the previous strategies for moving knots. Then in Section 3, the knots redistribution and PDE solutions are estimated step by step applying MQ quasi-interpolation. Numerical experiments are presented to illustrate the validity of the proposed method in Section 4. Section 5 ends the paper with some conclusions and remarks.

2. The new moving knots strategy based on EP

In this section, two classical moving knots strategies based on EP are recalled. On the basis of them, the new strategy for moving knots is developed.

We begin this section with the introduction of EP. Given a function $u = u(x,t)$, $\{x_j(t)\}_{j=0}^N$ are called equidistributed, if for fixed t, the arc-length of the curve $u(x, t)$ on each interval is the same:

$$\int_{x_j(t)}^{x_{j+1}(t)} M(\tilde{x},t)d\tilde{x} = \frac{1}{N}\int_0^1 M(\tilde{x},t)d\tilde{x}, \tag{2.1}$$

$M = \sqrt{1+u_x^2}$ is differentiation of the arc-length.

Denoting the error of the equidistribution as $E(x_j(t), t)$:

$$E(x_j(t),t) = \int_0^{x_j(t)} M(\tilde{x},t)d\tilde{x} - \frac{j}{N}\int_0^1 M(\tilde{x},t)d\tilde{x}, \tag{2.2}$$

then searching the roots of Eq. (2.1) is equivalent to finding the zero points of function (2.2).

When applying EP to construct the moving knots strategy, one aims to solve the exact zero points of function (2.2) [5]. However, The nonlinearity of function (2.2) makes the calculation difficult. Hence, a variety of techniques are established to simulate the zero points of function (2.2), for more information, we refer readers to the book [5] and the citations therein.

2.1. The classical moving knots strategies

In this subsection, two typical moving knots strategies are introduced: Huang's MMPDEs [7] and the moving knots equation in [10].

2.1.1. Huang's MMPDEs

Huang [7] gave the continuous form of the EP (2.3) by introducing a computational coordinate ξ. Then a one-to-one coordinate transformation is denoted as $x = x(\xi, t)$ with the boundary conditions $x(0, t) = 0$ and $x(1, t) = 1$.

$$E(x(\xi,t),t) = \int_0^{x(\xi,t)} M(\tilde{x},t)d\tilde{x} - \xi\int_0^1 M(\tilde{x},t)d\tilde{x} = 0. \tag{2.3}$$

Clearly, when $\xi_j = \frac{j}{N}, j = 0, 1, \dots, N$, the EP-contours $x_j(t) = x(\xi_j, t)$ are equidistributed on the surface $u(x, t)$.

On the basis of the continuous form of the EP (2.3), Huang [7] proposed a series of moving knots strategies named MMPDE 1–7 separately, the typical one is named MMPDE 5:

$$x_j^{k+1} = x_j^k - \frac{\Delta t_k}{\tau}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right), \tag{2.4}$$

where τ is a small constant and $\frac{\Delta t_k}{\tau}$ is the relaxation factor.

To facilitate a better understanding of MMPDE 5, we divide Eq. (2.4) into two iterations:

$$\begin{cases} x_j^{k,(1)} = x_j^k - \frac{\Delta t_k}{\tau}E_{\xi\xi}(x_j^k,t_k), & (2.5a) \\ x_j^{k+1} = x_j^{k,(1)}. & (2.5b) \end{cases}$$

$x_j^{k,(1)}$ is the intermediate term.

It is observed that, Eq. (2.5a) is taken one step of Iteration Method on $E_{\xi\xi} = 0$ at time t_k. Hence (2.5a) derives the knots $x_j^{k,(1)}$ closer to the equidistributed case on $u(x, t_k)$. Afterward, Eq. (2.5b) let x_j^{k+1} be equal to $x_j^{k,(1)}$ directly. This implies that, once the solution $u(x, t)$ varies largely from t_k to t_{k+1}, x_j^{k+1} may be even farther from the equidistributed knots on $u(x, t_{k+1})$. That is the reason why the time step sizes of Huang's MMPDEs [7] should be small, otherwise, instability and even knots tangling phenomena would emerge.